



\*\*FILE\*\* ID\*\*SUBMIT

110

```
1 0001 0 MODULE SUBMIT  (XTITLE 'PRINT and SUBMIT Commands'  
2 0002 0  
3 0003 0  
4 0004 0  
5 0005 1 BEGIN  
6 0006 1  
7 0007 1 *****  
8 0008 1 *  
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
11 0011 1 * ALL RIGHTS RESERVED.  
12 0012 1 *  
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
18 0018 1 * TRANSFERRED.  
19 0019 1 *  
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
22 0022 1 * CORPORATION.  
23 0023 1 *  
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
26 0026 1 *  
27 0027 1 *  
28 0028 1 *****  
29 0029 1  
30 0030 1  
31 0031 1 **  
32 0032 1 * FACILITY:  
33 0033 1 * PRINT and SUBMIT commands.  
34 0034 1  
35 0035 1 * ABSTRACT:  
36 0036 1 * This module implements the PRINT and SUBMIT commands.  
37 0037 1  
38 0038 1 * ENVIRONMENT:  
39 0039 1 * VAX/VMS user mode.  
40 0040 1  
41 0041 1 *--  
42 0042 1 * AUTHOR: M. Jack, CREATION DATE: 30-Apr-1982  
43 0043 1  
44 0044 1 * MODIFIED BY:  
45 0045 1 * V03-009 RRB0021 Rowland R. Bradley 27-Aug-1984  
46 0046 1 * Disallow the submission of command procedures and  
47 0047 1 * and print jobs from sequential devices. Avoids the  
48 0048 1 * problem described in Crash 201 (symbiont in RWAST state).  
49 0049 1 * We do not support sharing sequential devices (mag tape).  
50 0050 1 * Clean up the error logging associated with PRINT and  
51 0051 1 * SUBMIT.  
52 0052 1  
53 0053 1 * V03-008 JAK0220 J A Krycka 21-Jul-1984  
54 0054 1 * Make a SSNDJBC item list entry if /FEED is explicitly present.  
55 0055 1 * Before, an item list entry was made only for /NOFEED because  
56 0056 1 * /FEED was the default condition. Now the user can establish a  
57 0057 1 * queue level default value for this attribute, so it is
```

58 0058 1 | necessary to be able to override the queue default.  
59 0059 1 |  
60 0060 1 |  
61 0061 1 |  
62 0062 1 |  
63 0063 1 |  
64 0064 1 |  
65 0065 1 |  
66 0066 1 |  
67 0067 1 |  
68 0068 1 |  
69 0069 1 |  
70 0070 1 |  
71 0071 1 |  
72 0072 1 |  
73 0073 1 |  
74 0074 1 |  
75 0075 1 |  
76 0076 1 |  
77 0077 1 |  
78 0078 1 |  
79 0079 1 |  
80 0080 1 |  
81 0081 1 |  
82 0082 1 |  
83 0083 1 |  
84 0084 1 |  
85 0085 1 |  
86 0086 1 | \*\*  
V03-007 RRB0020 Rowland R. Bradley 18-Jun-1984  
Add support for common command qualifier processing  
This is a late V4 enhancement ask for by customers  
to allow dumb user file filtering.  
Add enhanced error message handling for /REMOTE  
and remove redundant error message when no files  
in the job. Add REMOINSF error code.  
V03-006 BLS0267 Benn Schreiber 31-Jan-1984  
Use new LIB\$FIND\_FILE argument.  
V03-005 MLJ0115 Martin L. Jack, 29-Jul-1983 13:40  
Changes for job controller baselevel.  
V03-004 MLJ0113 Martin L. Jack, 26-May-1983 17:00  
Changes for job controller baselevel.  
V03-003 MLJ0112 Martin L. Jack, 29-Apr-1983 0:17  
Changes for job controller baselevel.  
V03-002 MLJ0109 Martin L. Jack, 13-Apr-1983 14:44  
Disallow node name without /REMOTE.  
V03-001 MLJ0106 Martin L. Jack, 1-Mar-1983 5:03  
Changes for job controller baselevel.

```
88      0087 1 LIBRARY 'SYSSLIBRARY:STARLET';
89      0088 1 REQUIRE 'SRC$:JBCPRSDEF';
90      0198 1
91      0199 1
92      0200 1 LITERAL
93      0201 1      TRUE=      1;
94      0202 1      FALSE=     0;
95      0203 1
96      0204 1
97      0205 1 STRUCTURE
98      0206 1      BBLOCK[0,P,S,E;N]=
99      0207 1      [N]
100     0208 1      (BBLOCK + 0)<P,S,E>;
101     0209 1
102     0210 1
103     0211 1 PSECT
104     0212 1      CODE=      CODE.
105     0213 1      PLIT=      CODE;
106     0214 1      OWN=      DATA;
107     0215 1      GLOBAL=     DATA;
108     0216 1
109     0217 1
110     0218 1 FORWARD ROUTINE
111     0219 1      SUBMIT,
112     0220 1      GET COMM_QUAL_CONTEXT: NOVALUE,
113     0221 1      EXIT HANDLER: NOVALUE,
114     0222 1      HANDLER;
115     0223 1
116     0224 1
117     0225 1 EXTERNAL ROUTINE
118     0226 1      CLISGET_VALUE: ADDRESSING_MODE(GENERAL),
119     0227 1                      ! Get a parameter or qualifier value
120     0228 1      CLISPRESNT: ADDRESSING_MODE(GENERAL),
121     0229 1                      ! Determine if entity is present
122     0230 1      LIBSFIND_FILE: ADDRESSING_MODE(GENERAL),
123     0231 1                      ! Scan for wildcard files
124     0232 1      LIBSSCOPY_DDX: ADDRESSING_MODE(GENERAL),
125     0233 1                      ! Copy string
126     0234 1      LIBSQUAL_FILE_MATCH: ADDRESSING_MODE(GENERAL),
127     0235 1                      ! Match common qualifiers
128     0236 1      LIBSQUAL_FILE_PARSE: ADDRESSING_MODE(GENERAL);
129     0237 1                      ! Setup common qualifiers
130     0238 1
131     0239 1 EXTERNAL LITERAL
132     0240 1      ! Common Qualifier return status values.
133     0241 1
134     0242 1      LIBS QUIPRO.          ! Quit Processing
135     0243 1      LIBS NEGANS.        ! Negative Answer for this file
136     0244 1      LIBS FILFAIMAT.     ! File failed to meet criteria
137     0245 1
138     0246 1      ! Parallel PRINT and SUBMIT messages.
139     0247 1
140     0248 1      PRINTS FACILITY.    SUBMITS FACILITY.
141     0249 1      PRINTS ADDFIL.       SUBMITS ADDFIL.
142     0250 1      PRINTS CREJOB.      SUBMITS CREJOB.
143     0251 1      PRINTS INVFIL$PE.  SUBMITS INVFIL$PE.
144     0252 1      PRINTS REMOINSF.   SUBMITS REMOINSF.
```

SUBMIT  
V04-000

PRINT and SUBMIT Commands

M 10

16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55

VAX-11 Bliss-32 v4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 4  
(2)

```
: 145      0253 1
: 146      0254 1      ! SUBMIT only messages.
: 147      0255 1
: 148      0256 1      SUBMITS_INVLOGFIL;
: 149      0257 1
: 150      0258 1      GLOBAL
: 151      0259 1      QUAL_CONTEXT;          ! Context for Common Qualifiers
: 152      0260 1
: 153      0261 1      OWN
: 154      0262 1      EXIT_STATUS:  BBLOCK[4];
: 155      0263 1      FLAGS:      BBLOCK[4];      ! Most severe error status
: 156      0264 1
: 157      0265 1
: 158      0266 1      MACRO
: 159      0267 1      V_PRINT=      0,0,1,0 %;      ! PRINT command
: 160      0268 1      V_IDENTIFY=   0,1,1,0 %;      ! /IDENTIFY
: 161      0269 1      V_REMOTE=     0,2,1,0 %;      ! /REMOTE
: 162      0270 1      V_NO_LOG_FILE= 0,3,1,0 %;      ! /NOLOG_FILE
```

```

164 0271 1 ROUTINE SUBMIT=
165 0272 1
166 0273 1 ++
167 0274 1
168 0275 1 FUNCTIONAL DESCRIPTION:
169 0276 1 This routine is the main entry point for the PRINT and SUBMIT commands.
170 0277 1
171 0278 1 INPUT PARAMETERS:
172 0279 1 Standard VMS activation parameters (not used).
173 0280 1
174 0281 1 IMPLICIT INPUTS:
175 0282 1 NONE
176 0283 1
177 0284 1 OUTPUT PARAMETERS:
178 0285 1 NONE
179 0286 1
180 0287 1 IMPLICIT OUTPUTS:
181 0288 1 NONE
182 0289 1
183 0290 1 ROUTINE VALUE:
184 0291 1 Completion status.
185 0292 1
186 0293 1 SIDE EFFECTS:
187 0294 1 NONE
188 0295 1
189 0296 1 --
190 0297 1
191 0298 2 BEGIN LOCAL
192 0299 2
193 0300 2 CNT: INITIAL(0),
194 0301 2 FIND_CONTEXT: REF BBLOCK,
195 0302 2 FIND_RESULT: BBLOCK[DSC$C-S-BLN],
196 0303 2 FIND_DEFAULT: BBLOCK[DSC$C-S-BLN],
197 0304 2 VALUE_DESC: BBLOCK[DSC$C-S-BLN],
198 0305 2 INPUT_DESC: BBLOCK[DSC$C-S-BLN],
199 0306 2 LOG_FILE_DESC: BBLOCK[DSC$C-S-BLN],
200 0307 2 JSM_BUFFER: VECTOR[132,BYTE],
201 0308 2 JSM_DESC: VECTOR[2],
202 0309 2 ITEM_BUFFER: BBLOCK[1024],
203 0310 2 DATA_BUFFER: BBLOCK[1024],
204 0311 2 EXIT_BLOCK: VECTOR[5];
205 0312 2
206 0313 2 MACRO
207 P 0314 2 SD_[A] = BIND %NAME('D_', A) = $DESCRIPTOR(A) %;
208 P 0315 2 SD_(
209 P 0316 2   'SVERB',
210 P 0317 2   'P1',
211 P 0318 2   'AFTER',
212 P 0319 2   'BURST',
213 P 0320 2   'CHARACTERISTICS',
214 P 0321 2   'CLI',
215 P 0322 2   'COPIES',
216 P 0323 2   'CPUTIME',
217 P 0324 2   'DELETE',
218 P 0325 2   'DEVICE',
219 P 0326 2   'FEED',
220 P 0327 2   'FLAG',
221 P 0328 2   'FORM',
222 P 0329 2

```

```
221      P 0328 2      'HEADER',  
222      P 0329 2      'HOLD',  
223      P 0330 2      'IDENTIFY',  
224      P 0331 2      'JOB COUNT',  
225      P 0332 2      'KEEP',  
226      P 0333 2      'LOG FILE',  
227      P 0334 2      'LOWERCASE',  
228      P 0335 2      'NAME',  
229      P 0336 2      'NOTE',  
230      P 0337 2      'NOTIFY',  
231      P 0338 2      'OPERATOR',  
232      P 0339 2      'PAGES',  
233      P 0340 2      'PARAMETERS',  
234      P 0341 2      'PASSALL',  
235      P 0342 2      'PRINTER',  
236      P 0343 2      'PRIORITY',  
237      P 0344 2      'QUEUE',  
238      P 0345 2      'REMOTE',  
239      P 0346 2      'RESTART',  
240      P 0347 2      'SETUP',  
241      P 0348 2      'SPACE',  
242      P 0349 2      'TRAILER',  
243      P 0350 2      'USER',  
244      P 0351 2      'WSDEFAULT',  
245      P 0352 2      'WSEXTENT',  
246      P 0353 2      'WSQUOTA');  
247      0354 2      BIND  
248      0355 2      D_BURST_ALL= $DESCRIPTOR('BURST.ALL'),  
249      0356 2      D_BURST_ONE= $DESCRIPTOR('BURST.ONE'),  
250      0357 2      D_FLAG_ALL= $DESCRIPTOR('FLAG.ALL'),  
251      0358 2      D_FLAG_ONE= $DESCRIPTOR('FLAG.ONE'),  
252      0359 2      D_TRAILER_ALL= $DESCRIPTOR('TRAILER.ALL'),  
253      0360 2      D_TRAILER_ONE= $DESCRIPTOR('TRAILER.ONE');  
254      0361 2      BUILTIN  
255      0362 2      FP;  
256      0363 2      PARSE_GLOBAL_REGISTERS;  
257      0364 2  
258      0365 2  
259      0366 2  
260      0367 2      ! Initialize.  
261      0368 2  
262      0369 2      EXIT_STATUS = SSS_NORMAL;  
263      0370 2      .FP = HANDLER;  
264      0371 2  
265      0372 2      INPUT_DESC[DSCSB_CLASS] = DSCSK_CLASS_D;  
266      0373 2      INPUT_DESC[DSCSB_DTYPE] = DSCSK_DTYPE_T;  
267      0374 2      INPUT_DESC[DSCSW_LENGTH] = 0;  
268      0375 2      INPUT_DESC[DSCSA_POINTER] = 0;  
269      0376 2  
270      0377 2      VALUE_DESC[DSCSB_CLASS] = DSCSK_CLASS_D;  
271      0378 2      VALUE_DESC[DSCSB_DTYPE] = DSCSK_DTYPE_T;  
272      0379 2      VALUE_DESC[DSCSW_LENGTH] = 0;  
273      0380 2      VALUE_DESC[DSCSA_POINTER] = 0;  
274      0381 2  
275      0382 2      LOG_FILE_DESC[DSCSB_CLASS] = DSCSK_CLASS_D;  
276      0383 2      LOG_FILE_DESC[DSCSB_DTYPE] = DSCSK_DTYPE_T;  
277      0384 2      LOG_FILE_DESC[DSCSW_LENGTH] = 0;
```

```
278 0385 2 LOG_FILE_DESC[DSCSA_POINTER] = 0;
279 0386 2
280 0387 2 FIND_RESULT[DSC$B_CLASS] = DSCSK_CLASS_D;
281 0388 2 FIND_RESULT[DSC$B_DTYPE] = DSCSK_DTYPE_T;
282 0389 2 FIND_RESULT[DSC$W_LENGTH] = 0;
283 0390 2 FIND_RESULT[DSCSA_POINTER] = 0;
284 0391 2
285 0392 2 FIND_CONTEXT = 0;
286 0393 2
287 0394 2 FIND_DEFAULT[DSC$B_CLASS] = DSCSK_CLASS_S;
288 0395 2 FIND_DEFAULT[DSC$B_DTYPE] = DSCSK_DTYPE_T;
289 0396 2 FIND_DEFAULT[DSC$W_LENGTH] = %CHARCOUNT('.COM');
290 0397 2 FIND_DEFAULT[DSCSA_POINTER] = UPLIT BYTE('.COM');
291 0398 2
292 0399 2 FLAGS = 0;
293 0400 2
294 0401 2
295 0402 2 ! Initialize for command parsing utilities.
296 0403 2
297 P 0404 2 PARSE_GLOBAL_INIT(
298 P 0405 2     I_CURSOR= ITEM_BUFFER,
299 P 0406 2     DCURSOR= DATA_BUFFER,
300 P 0407 2     MESSAGE= SUBMITS FACILITY^16 + SHRS_INVQUAVAL + STSSK_SEVERE,
301 0408 2     VALUE_DESC= VALUE_DESC);
302 0409 2
303 0410 2
304 0411 2 ! Get the command verb. If it is PRINT, rather than SUBMIT, change some
305 0412 2 defaults.
306 0413 2
307 0414 2 CLISGET VALUE(D $VERB, VALUE_DESC);
308 0415 2 IF CH$RCHAR(.VALUE_DESC[DSCSA_POINTER]) EQL XC'P'
309 0416 2 THEN
310 0417 2     BEGIN
311 0418 2         FLAGS[V_PRINT] = TRUE;
312 0419 2         FIND_DEFAULT[DSCSA_POINTER] = UPLIT BYTE ('.LIS');
313 0420 2     END;
314 0421 2
315 0422 2 ! Get the common qualifiers context for the PRINT command
316 0423 2
317 0424 2 GET_COMM_QUAL_CONTEXT();
318 0425 2
319 0426 2
320 0427 2 ! Get the command qualifiers.
321 0428 2
322 0429 2 FLAGS[V_IDENTIFY] = CLISPRESENT(D IDENTIFY);
323 0430 2 FLAGS[V_REMOTE] = CLISPRESENT(D_REMOTE);
324 0431 2
325 0432 2
326 0433 2 ! Get the /AFTER qualifier.
327 0434 2
328 0435 2 PARSE_CALL(AFTER, D_AFTER);
329 0436 2
330 0437 2
331 0438 2 ! Get the /CHARACTERISTICS qualifier.
332 0439 2
333 0440 2 PARSE_CALL(CHARACTERISTICS, D_CHARACTERISTICS);
334 0441 2
```

```
335      0442 2
336      0443 2 ! Get the /HOLD qualifier.
337      0444 2
338      P 0445 2 PARSE_CALL(IF_TRUE, D_HOLD,
339          0446 2 SJCS_HOLD);
340          0447 2
341          0448 2
342          0449 2 ! Get the /NOTIFY qualifier.
343          0450 2
344          P 0451 2 PARSE_CALL(IF_TRUE, D_NOTIFY,
345              0452 2 SJCS_NOTIFY);
346              0453 2
347              0454 2
348              0455 2 ! Get the /PARAMETERS qualifier.
349              0456 2
350              0457 2 PARSE_CALL(PARAMETERS, D_PARAMETERS);
351              0458 2
352              0459 2
353              0460 2 ! Get the /PRIORITY qualifier.
354              0461 2
355              0462 2 PARSE_CALL(PRIORITY, D_PRIORITY);
356              0463 2
357              0464 2
358              0465 2 ! Get the /RESTART qualifier.
359              0466 2
360              P 0467 2 PARSE_CALL(IF_TRUE FALSE, D_RESTART,
361                  0468 2 SJCS_RESTART, SJCS_NO_RESTART);
362                  0469 2
363                  0470 2
364                  0471 2 ! Get the /USER qualifier.
365                  0472 2
366                  0473 2 PARSE_CALL(USER, D_USER);
367                  0474 2
368                  0475 2
369                  0476 2 IF .FLAG$[V_PRINT]
370                  0477 2 THEN
371                      0478 2 BEGIN
372                          0479 2
373                          0480 2 ! Get the command-level /BURST qualifier.
374                          0481 2
375                          0482 2 PARSE_CALL(IF_TRUE, D_BURST_ALL, SJCS_FILE_BURST);
376                          0483 2 PARSE_CALL(IF_TRUE, D_BURST_ONE, SJCS_FILE_BURST_ONE);
377                          0484 2 PARSE_CALL(IF_FALSE, D_BURST, SJCS_NO_FILE_BURST);
378                          0485 2
379                          0486 2
380                          0487 2 ! Get the /DEVICE qualifier.
381                          0488 2
382                          0489 2 IF CLIPRESENT(D_DEVICE) AND NOT CLIPRESENT(D_QUEUE)
383                          0490 2 THEN
384                              0491 2 PARSE_CALL(QUEUE, D_DEVICE, SJCS_QUEUE, $DESCRIPTOR('SY$PRINT'));
385                              0492 2
386                              0493 2
387                              0494 2 ! Get the command-level /FLAG qualifier.
388                              0495 2
389                              0496 2 PARSE_CALL(IF_TRUE, D_FLAG_ALL, SJCS_FILE_FLAG);
390                              0497 2 PARSE_CALL(IF_TRUE, D_FLAG_ONE, SJCS_FILE_FLAG_ONE);
391                              0498 2 PARSE_CALL(IF_FALSE, D_FLAG, SJCS_NO_FILE_FLAG);
```



```
449      0556 3
450      0557
451      0558
452      P 0559
453      0560
454      0561
455      0562
456      0563
457      P 0564
458      0565
459      0566
460      0567
461      0568
462      0569
463      0570
464      0571
465      0572
466      0573
467      0574
468      0575
469      0576
470      0577
471      0578
472      0579
473      0580
474      0581
475      0582
476      0583
477      0584
478      P 0585
479      P 0586
480      0587
481      0588
482      0589
483      0590
484      0591
485      0592
486      0593
487      0594
488      0595
489      0596
490      0597
491      0598
492      0599
493      0600
494      P 0601
495      0602
496      0603
497      0604
498      0605
499      0606
500      P 0607
501      0608
502      0609
503      0610
504      0611
505      0612

      ! Get the /CPUETIME qualifier.
      PARSE_CALL(CPUTIME, D_CPUTIME,
                 SJCS_CPU_LIMIT, SJCS_NO_CPU_LIMIT);

      ! Get the /KEEP qualifier.
      PARSE_CALL(IF_TRUE, D_KEEP,
                 SJCS_NO_LOG_DELETE);

      ! Get the /LOG_FILE qualifier.
      Q VALUE_DESC = LOG_FILE_DESC;
      FLAG[V_NO_LOG_FILE] = PARSE_CALL_VALUE(LOG_FILE, D_LOG_FILE);

      ! Get the /NAME qualifier.
      Q VALUE_DESC = VALUE_DESC;
      PARSE_CALL(NAME, D_NAME);

      ! Get the log file specification.
      IF NOT .FLAG[V_NO_LOG_FILE]
      THEN
          PARSE_CALL(NAME_AND_LOG_FILE,
                     VALUE_DESC, LOG_FILE_DESC,
                     $DESCRIPTOR('SYSSLOGIN:.LOG'), SUBMITS_INVLOGFL);

      ! Get the /PRINTER qualifier.
      PARSE_CALL(PRINTER, D_PRINTER);

      ! Get the /QUEUE qualifier.
      PARSE_CALL(QUEUE, D_QUEUE, SJCS_QUEUE, $DESCRIPTOR('SYSSBATCH'));

      ! Get the /WSDEFAULT qualifier.
      PARSE_CALL(WORKING_SET, D_WSDEFAULT,
                 SJCS_WSDEFAULT, SJCS_NO_WSDEFAULT);

      ! Get the /WSEXTENT qualifier.
      PARSE_CALL(WORKING_SET, D_WSEXTENT,
                 SJCS_WSEXTENT, SJCS_NO_WSEXTENT);

      ! Get the /WSQUOTA qualifier.
```

```
506      0613 3      !  
507      0614      PARSE_CALL(WORKING_SET, D_WSQUOTA,  
508      0615      SJCS_WSQUOTA, SJCS_NO_WSQUOTAS);  
509      0616      END;  
510      0617  
511      0618  
512      0619      ! Create the job.  
513      0620      IF NOT .FLAG$[V_REMOTE]  
514      0621      THEN  
515      0622      BEGIN  
516      0623      LOCAL  
517      0624      IOSB:      VECTOR[4,WORD], ! I/O status block  
518      0625      STATUS:      ! Status return  
519      0626  
520      0627  
521      0628  
522      0629  
523      0630  
524      0631      ! Finish the list and send the create job request.  
525      0632      Q_ICURSOR[0,0,32,0] = 0;  
526      0633      STATUS = $SNDJBCW(  
527      0634      FUNC=SJCS_CREATE_JOB,  
528      0635      ITMLST=ITEM_BUFFER,  
529      0636      IOSB=IOSB);  
530      0637      IF .STATUS THEN STATUS = .IOSB;  
531      0638      IF NOT .STATUS  
532      0639      THEN  
533      0640      SIGNAL STOP(  
534      0641      SUBMITS_CREJOB, 0, .STATUS);  
535      0642  
536      0643  
537      0644      ! Establish an exit handler to delete the open job on an abnormal exit.  
538      0645      EXIT_BLOCK[0] = 0;  
539      0646      EXIT_BLOCK[1] = EXIT_HANDLER;  
540      0647      EXIT_BLOCK[2] = 1;  
541      0648      EXIT_BLOCK[3] = EXIT_BLOCK[4];  
542      0649      EXIT_BLOCK[4] = 0;  
543      0650      $DCLEXH(DESBLK=EXIT_BLOCK);  
544      0651      END;  
545      0652  
546      0653  
547      0654      ! Loop for all elements of the parameter list.  
548      0655      WHILE CLISGET_VALUE(D_P1, INPUT_DESC) DO  
549      0656      BEGIN  
550      0657      LOCAL  
551      0658      STATUS_3,  
552      0659      STATUS_2;  
553      0660  
554      0661  
555      0662      ! Set up the message buffer for the ADDFIL request.  
556      0663  
557      0664      Q_ICURSOR = ITEM_BUFFER;  
558      0665      Q_DCURSOR = DATA_BUFFER;  
559      0666  
560      0667  
561      0668      ! Get the /DELETE qualifier.  
562      0669
```

```
563 P 0670 3 PARSE_CALL(IF_TRUE, D_DELETE,  
564 0671 SJCS_DELETE_FILE);  
565 0672  
566 0673  
567 0674 IF_FLAGS[V_PRINT]  
568 0675 THEN BEGIN  
569 0676  
570 0677  
571 0678  
572 0679  
573 P 0680  
574 0681 PARSE_CALL(LOCAL TRUE FALSE, D_BURST,  
575 0682 SJCS_FILE_BURST, SJCS_NO_FILE_BURST);  
576 0683  
577 0684  
578 0685  
579 P 0686  
580 0687 PARSE_CALL(COPIES, D_COPIES,  
581 0688 SJCS_FILE_COPIES);  
582 0689  
583 0690  
584 0691  
585 P 0692  
586 0693 PARSE_CALL(IF_TRUE FALSE, D_FEED,  
587 0694 SJCS_PAGIRATE, SJCS_NO_PAGINATE);  
588 0695  
589 0696  
590 0697  
591 P 0698  
592 0699 PARSE_CALL(LOCAL TRUE FALSE, D_FLAG,  
593 0700 SJCS_FILE_FLAG, SJCS_NO_FILE_FLAG);  
594 0701  
595 0702  
596 0703  
597 P 0704  
598 0705 PARSE_CALL(IF_TRUE, D_HEADER,  
599 0706 SJCS_PAGE_HEADER);  
600 0707  
601 0708  
602 0709  
603 P 0710  
604 P 0711 PARSE_CALL(LOWER UPPER, D_PAGES,  
605 0712 SJCS_FIRST_PAGE, SJCS_NO_FIRST_PAGE,  
606 0713 SJCS_LAST_PAGE, SJCS_NO_LAST_PAGE);  
607 0714  
608 0715  
609 0716  
610 P 0717  
611 0718 PARSE_CALL(IF_TRUE, D_PASSALL,  
612 0719 SJCS_PASSALL);  
613 0720  
614 0721  
615 0722  
616 P 0723  
617 0724 PARSE_CALL(COMMA_LIST, D_SETUP,  
618 0725 SJCS_FILE_SETUP_MODULES, SJCS_NO_FILE_SETUP_MODULES);  
619 0726
```

```

620      0727 4      ! Get the /SPACE qualifier.
621      0728 4
622      P 0729 4      PARSE_CALL(IF_TRUE, D_SPACE,
623          0730 4      SJCS_DOUBLE_SPACES);
624          0731 4
625          0732 4
626          0733 4      ! Get the local-level /TRAILER qualifier.
627          P 0734 4      PARSE_CALL(LOCAL_TRUE_FALSE, D_TRAILER,
628              0735 4      SJCS_FILE_TRAILER, SJCS_NO_FILE_TRAILER);
629              0736 4
630              0737 4      END:
631          0738 4
632          0739 4
633          0740 4      ! Loop, calling LIBSFIND_FILE to get files matching the input spec.
634          0741 4
635          UNTIL
636          0742 4      BEGIN
637          0743 4      STATUS_2 = LIBSFIND_FILE(
638              0744 4      INPUT_DESC,
639              0745 4      FIND_RESULT,
640              0746 4      FIND_CONTEXT,
641              0747 4      FIND_DEFAULT,
642              0748 4      0,0,%REF(2));
643              0749 4      .STATUS_2 EQL RMSS_NMF
644          0750 4      !Request multiple input file processing
645          END
646          DO
647          BEGIN
648          IF NOT .STATUS_2
649          THEN
650          BEGIN
651          SIGNAL(
652              (IF .FLAGS[V PRINT]
653                  THEN SUBMITS_FACILITY^16 + SHRS_OPENIN + STSSK_ERROR
654                  ELSE SUBMITS_FACILITY^16 + SHRS_OPENIN + STSSK_SEVERE),
655              1, FIND_RESULT,
656              .FIND_CONTEXT[FABSL_STS], .FIND_CONTEXT[FABSL_STV]);
657          END
658          ELSE
659          BEGIN
660          LOCAL
661          PROMPT_DESC;
662          PROMPT_DESC = FIND_RESULT;
663          ! Pointer to a desc of
664          ! the filename
665          STATUS_3 = LIBSQUAL FILE_MATCH (
666              QUAL_CONTEXT,
667              FIND_CONTEXT,
668              0,
669              (IF .FLAGS[V PRINT]
670                  THEN $DESCRIPTOR('!AS, Print? [N]:')
671                  ELSE $DESCRIPTOR('!AS, Submit? [N]:')),
672                  PROMPT_DESC,
673                  0);
674          ! Prompt
675          ! Descriptor filename
676          ! No prompt routine
677          IF NOT (.STATUS_3)
678
679
680
681
682
683

```

```
677      0784 5      THEN
678      0785 6      BEGIN
679      0786 6      IF (.STATUS_3 NEQ LIBS_FILFAIMAT) AND
680      0787 6      (.STATUS_3 NEQ LIBS QUIPRO) AND
681      0788 7      (.STATUS_3 NEQ LIBS_NEGANS)
682      0789 6      THEN
683      0790 7      BEGIN
684      0791 7      SIGNAL (SUBMITS FACILITY*16 +
685      0792 7      SHRS_F[NOTACC + STSSK_WARNING,
686      0793 7      1, FIND_RESULT);
687      0794 7      END
688      0795 7      ELSE IF (.STATUS_3 EQ LIBS QUIPRO)
689      0796 6      THEN
690      0797 6      EXITLOOP;          ! Quit Now
691      0798 6      END
692      0799 5      ELSE
693      0800 6      BEGIN
694      0801 6      FIND_CONTEXT[FABSV_SPL] = FIND_CONTEXT[FABSV_SCF] = FALSE;
695      0802 6
696      0803 6
697      0804 7
698      0805 6
699      0806 6
700      0807 7
701      0808 7
702      0809 6
703      0810 6
704      0811 6
705      0812 6
706      0813 7
707      0814 7
708      0815 7
709      0816 8
710      0817 8
711      0818 8
712      0819 8
713      0820 8
714      0821 8
715      0822 8
716      0823 8
717      0824 8
718      0825 8
719      0826 8
720      0827 7
721      0828 7
722      0829 8
723      0830 7
724      0831 7
725      0832 7
726      0833 7
727      0834 7
728      0835 7
729      0836 7
730      0837 7
731      0838 7
732      0839 8
733      0840 8

      THEN
      BEGIN
      IF (.STATUS_3 NEQ LIBS_FILFAIMAT) AND
      (.STATUS_3 NEQ LIBS QUIPRO) AND
      (.STATUS_3 NEQ LIBS_NEGANS)
      THEN
      BEGIN
      SIGNAL (SUBMITS FACILITY*16 +
      SHRS_F[NOTACC + STSSK_WARNING,
      1, FIND_RESULT);
      END
      ELSE IF (.STATUS_3 EQ LIBS QUIPRO)
      THEN
      EXITLOOP;          ! Quit Now
      END
      BEGIN
      FIND_CONTEXT[FABSV_SPL] = FIND_CONTEXT[FABSV_SCF] = FALSE;
      IF NOT SOPEN(FAB=.FIND_CONTEXT)
      THEN
      SIGNAL(
      (IF .FLAGS[V_PRINT]
      THEN SUBMITS FACILITY*16 + SHRS_OPENIN + STSSK_ERROR
      ELSE SUBMITS FACILITY*16 + SHRS_OPENIN + STSSK_SEVERE),
      1, FIND_RESULT,
      .FIND_CONTEXT[FABSL_STS], .FIND_CONTEXT[FABSL_STV])
      ELSE
      BEGIN
      IF .FLAGS[V_REMOTE]
      THEN
      BEGIN
      IF NOT .BBLOCK[FIND_CONTEXT[FABSL_DEV], DEV$V_NET]
      THEN
      SIGNAL(
      SUBMITS_REMOTEINSF,
      1, FIND_RESULT)
      ELSE
      IF .FLAGS[V_PRINT]
      THEN FIND_CONTEXT[FABSV_SPL] = TRUE
      ELSE FIND_CONTEXT[FABSV_SCF] = TRUE;
      END;
      IF NOT SCLOSE(FAB=.FIND_CONTEXT)
      THEN
      SIGNAL(
      SUBMITS FACILITY*16 + SHRS_CLOSEIN + STSSK_ERROR,
      1, FIND_RESULT,
      .FIND_CONTEXT[FABSL_STS], .FIND_CONTEXT[FABSL_STV]);
      IF NOT .FLAGS[V_REMOTE]
      THEN
      BEGIN
      IF .BBLOCK[FIND_CONTEXT[FABSL_DEV], DEV$V_NET]
```



```

791 0898 3 BEGIN
792 0899 LOCAL
793 0900 IOSB: VECTOR[4,WORD], | I/O status block
794 0901 STATUS: | Status return
795 0902
796 0903
797 0904 ! Add the JOB_STATUS_OUTPUT item and finish the list.
798 0905
799 0906 Q_ICURSOR = ITEM_BUFFER;
800 0907 IF .FLAGS[V_IDENTITY]
801 0908 THEN
802 0909 BEGIN
803 0910 JSM_DESC[0] = 0;
804 0911 JSM_DESC[1] = JSM_BUFFER;
805 0912 Q_ICURSOR[0,0,16,0] = XALLOCATION(JSM_BUFFER);
806 0913 Q_ICURSOR[2,0,16,0] = SJCS_JOB_STATUS_OUTPUT;
807 0914 Q_ICURSOR[4,0,32,0] = JSM_BUFFER;
808 0915 Q_ICURSOR[8,0,32,0] = JSM_DESC;
809 0916 Q_ICURSOR = Q_ICURSOR + T2;
810 0917 END;
811 0918 Q_ICURSOR[0,0,32,0] = 0;
812 0919
813 0920
814 0921 ! Send the close job request.
815 0922
816 0923 P STATUS = $SNDJBCW(
817 0924 FUNC=SJCS_CLOSE_JOB,
818 0925 ITMLST=ITEM_BUFFER,
819 0926 IOSB=IOSB);
820 0927
821 0928
822 0929
823 0930
824 0931 ! Cancel the exit handler established to delete the job on abnormal exit.
825 0932 SCANEXH(DESBLK=EXIT_BLOCK);
826 0933
827 0934 ! If close job failed and the count of jobs sent indicates that there
828 0935 WERE jobs then issue error.
829 0936
830 0937 IF .STATUS THEN STATUS = .IOSB;
831 0938 IF NOT .STATUS AND (.CNT NEQ 0)
832 0939 THEN
833 0940 SIGNAL STOP(
834 0941 SUBMITS_CREJOB, 0, .STATUS);
835 0942
836 0943
837 0944 ! Execute /IDENTIFY only if there WAS a job. Assume an error
838 0945 was issued if no tasks in the job.
839 0946
840 0947 IF .FLAGS[V_IDENTITY] AND (.CNT NEQ 0)
841 0948 THEN
842 0949 SIGNAL(
843 0950 SUBMITS_FACILITY^16 + SHRS_TEXT + STSSK_SUCCESS,
844 0951 1 OR XB^0001^16,
845 0952 JSM_DESC);
846 0953
847 0954 END;

```

SUBMIT  
V04-000

PRINT and SUBMIT Commands

M 11  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55  
VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 17  
(3)

: 848 0955 2  
: 849 0956 2 ! Return with the most severe error encountered.  
: 850 0957 2  
: 851 0958 2 ! EXIT\_STATUS OR STSSM\_INHIB\_MSG  
: 852 0959 1 END:

.TITLE SUBMIT PRINT and SUBMIT Commands  
.IDENT \V04-000\

.PSECT DATA.NOEXE.2

00000 QUAL\_CONTEXT::

00004 EXIT\_STATUS: .BLKB 4

00008 FLAGS: .BLKB 4

.PSECT CODE.NOWRT.2

42 52 45 56 24 00000 P.AAB: .ASCII \\$VERB\

00000005 00008 P.AAA: .BLKB 3

00000000 0000C P.AAD: .LONG 5

31 50 00010 P.AAD: .ADDRESS P.AAB

000012 .BLKB 4

00000002 00014 P.AAC: .LONG 2

00000000 00018 P.AAF: .ADDRESS P.AAD

52 45 54 46 41 0001C P.AAF: .ASCII \AFTER\

000021 .BLKB 3

00000005 00024 P.AAE: .LONG 5

00000000 00028 P.AAH: .ADDRESS P.AAF

54 53 52 55 42 0002C P.AAH: .ASCII \BURST\

000031 .BLKB 3

00000005 00034 P.AAG: .LONG 5

00000000 00038 P.AAJ: .ADDRESS P.AAH

53 43 49 54 53 49 52 45 54 43 41 52 41 48 43 0003C P.AAJ: .ASCII \CHARACTERISTICS\

00004B .BLKB 1

0000000F 0004C P.AAI: .LONG 15

00000000 00050 P.AAI: .ADDRESS P.AAJ

49 4C 43 00054 P.AAL: .ASCII \CLI\

000057 .BLKB 1

00000003 00058 P.AAK: .LONG 3

00000000 0005C P.AAN: .ADDRESS P.AAL

53 45 49 50 4F 43 00060 P.AAN: .ASCII \COPIES\

000066 .BLKB 2

00000006 00068 P.AAM: .LONG 6

00000000 0006C P.AAM: .ADDRESS P.AAN

45 4D 49 54 55 50 43 00070 P.AAP: .ASCII \CPUTIME\

000077 .BLKB 1

00000007 00078 P.AAO: .LONG 7

00000000 0007C P.AAO: .ADDRESS P.AAP

45 54 45 4C 45 44 00080 P.AAR: .ASCII \DELETE\

000086 .BLKB 2

00000006 00088 P.AAQ: .LONG 6

00000000 0008C P.AAQ: .ADDRESS P.AAR

45 43 49 56 45 44 00090 P.AAT: .ASCII \DEVICE\

SUBMIT  
V04-000

## PRINT and SUBMIT Commands

N 11  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55  
VAX-11 BLiss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1Page 18  
(3)

				00096		.BLKB 2
				00000006	00098	P.AAS: .LONG 6
				00000000	0009C	.ADDRESS P.AAT
44	45	45	46	000A0	000A0	P.AAV: .ASCII \FEED\
				00000004	000A4	P.AAU: .LONG 4
				00000000	000A8	.ADDRESS P.AAV
47	41	4C	46	000AC	000AC	P.AAX: .ASCII \FLAG\
				00000004	000B0	P.AAW: .LONG 4
				00000000	000B4	.ADDRESS P.AAX
4D	52	4F	46	000B8	000B8	P.AAZ: .ASCII \FORM\
				00000004	000BC	P.AAY: .LONG 4
				00000000	000C0	.ADDRESS P.AAZ
52	45	44	41	45	48	P.ABB: .ASCII \HEADER\
				00000006	000CA	.BLKB 2
				00000000	000CC	P.ABA: .LONG 6
				00000004	000D0	.ADDRESS P.ABB
44	4C	4F	48	000D4	000D4	P.ABD: .ASCII \HOLD\
				00000004	000D8	P.ABC: .LONG 4
				00000000	000DC	.ADDRESS P.ABD
59	46	49	54	4E	45	P.ABF: .ASCII \IDENTIFY\
				00000008	000E0	P.ABE: .LONG 8
				00000000	000E8	.ADDRESS P.ABF
54	4E	55	4F	43	5F	P.ABH: .ASCII \JOB_COUNT\
				00000009	000F0	.BLKB 3
				00000000	000F9	P.ABG: .LONG 9
				00000000	00100	.ADDRESS P.ABH
50	45	45	4B	00104	00104	P.ABJ: .ASCII \KEEP\
				00000004	00108	P.ABI: .LONG 4
				00000000	0010C	.ADDRESS P.ABJ
45	4C	49	46	5F	47	P.ABL: .ASCII \LOG_FILE\
				00000008	00110	P.ABK: .LONG 8
				00000000	0011C	.ADDRESS P.ABL
45	53	41	43	52	45	P.ABN: .ASCII \LOWERCASE\
				00000009	00120	.BLKB 3
				00000000	00129	P.ABM: .LONG 9
				00000000	00130	.ADDRESS P.ABN
45	4D	41	4E	00134	00134	P.ABP: .ASCII \NAME\
				00000004	00138	P.ABO: .LONG 4
				00000000	0013C	.ADDRESS P.ABP
45	54	4F	4E	00140	00140	P.ABR: .ASCII \NOTE\
				00000004	00144	P.ABQ: .LONG 4
				00000000	00148	.ADDRESS P.ABR
59	46	49	54	4F	4E	P.ABT: .ASCII \NOTIFY\
				00000006	0014C	.BLKB 2
				00000000	00152	P.ABS: .LONG 6
				00000006	00154	.ADDRESS P.ABT
52	4F	54	41	52	45	P.ABV: .ASCII \OPERATOR\
				00000008	0015C	P.ABU: .LONG 8
				00000000	00164	.ADDRESS P.ABV
53	45	47	41	50	0016C	P.ABX: .ASCII \PAGES\
				00000005	00171	.BLKB 3
				00000000	00174	P.ABW: .LONG 5
				00000000	00178	.ADDRESS P.ABX
53	52	45	54	45	4D	P.ABZ: .ASCII \PARAMETERS\
				0017C	00186	.BLKB 2
				0000000A	00188	P.ABY: .LONG 10
				00000000	0018C	.ADDRESS P.ABZ

4C 4C 41 53 53 41 50	00190	P.ACB:	.ASCII \PASSALL\	:
	00197		.BLKB 1	
	00000007	P.ACA:	.LONG 7	
	00000000	0019C	.ADDRESS P.ACB	
52 45 54 4E 49 52 50	001A0	P.ACD:	.ASCII \PRINTER\	
	001A7		.BLKB 1	
	00000007	P.ACC:	.LONG 7	
	00000000	001AC	.ADDRESS P.ACD	
59 54 49 52 4F 49 52 50	001B0	P.ACF:	.ASCII \PRIORITY\	
	001B8	P.ACE:	.LONG 8	
	00000008	001BC	.ADDRESS P.ACF	
	00000000	001C0	.ASCII \QUEUE\	
45 55 45 55 51	001C5	P.ACH:	.BLKB 3	
	001C8	P.ACG:	.LONG 5	
	00000005	001CC	.ADDRESS P.ACH	
45 54 4F 4D 45 52	001D0	P.ACJ:	.ASCII \REMOTE\	
	001D6		.BLKB 2	
	00000006	P.ACJ:	.LONG 6	
	00000000	001DC	.ADDRESS P.ACJ	
54 52 41 54 53 45 52	001E0	P.ACJ:	.ASCII \RESTART\	
	001E7	P.ACK:	.BLKB 1	
	00000007	001E8	.LONG 7	
	00000000	001EC	.ADDRESS P.ACJ	
50 55 54 45 53	001F0	P.ACN:	.ASCII \SETUP\	
	001F5		.BLKB 3	
	00000005	P.ACN:	.LONG 5	
	00000000	001FC	.ADDRESS P.ACN	
45 43 41 50 53	00200	P.ACN:	.ASCII \SPACE\	
	00205		.BLKB 3	
	00000005	P.ACN:	.LONG 5	
	00000000	0020C	.ADDRESS P.ACN	
52 45 4C 49 41 52 54	00210	P.ACR:	.ASCII \TRAILER\	
	00217	P.ACQ:	.BLKB 1	
	00000007	00218	.LONG 7	
	00000000	0021C	.ADDRESS P.ACR	
52 45 53 55	00220	P.ACT:	.ASCII \USER\	
	00224	P.ACS:	.LONG 4	
	00000004	00228	.ADDRESS P.ACT	
	00000000	0022C	.ASCII \WSDEFAULT\	
54 4C 55 41 46 45 44 53 57	00235	P.ACV:	.BLKB 3	
	00238	P.ACU:	.LONG 9	
	00000009	0023C	.ADDRESS P.ACV	
	00000000	00240	.ASCII \WSEXTENT\	
54 4E 45 54 58 45 53 57	00248	P.ACX:	.LONG 8	
	0024C	P.ACW:	.ADDRESS P.ACX	
	00000008	00250	.ASCII \WSQUOTA\	
	00000000	00257	.BLKB 1	
41 54 4F 55 51 53 57	00258	PACY:	.LONG 7	
	0025C	P.ACZ:	.ADDRESS P.ACZ	
	00000007	00260	.ASCII \BURST.ALL\	
4C 4C 41 2E 54 53 52 55 42	00269		.BLKB 3	
	0026C	P.ADA:	.LONG 9	
	00000009	00270	.ADDRESS P.ADB	
45 4E 4F 2E 54 53 52 55 42	00274	P.ADD:	.ASCII \BURST.ONE\	
	0027D		.BLKB 3	
	00000009	P.ADC:	.LONG 9	
	00000000	00284	.ADDRESS P.ADD	

SUBMIT  
V04-000

## PRINT and SUBMIT Commands

C 12  
16-Sep-1984 00:28:51 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:09:55 [CLIUTL.SRC]SUBMIT.B32;1Page 20  
(3)

4C 4C 41 2E 47 41 4C 46	00288 P.ADF: .ASCII \FLAG.ALL\
00000008 00000000	00290 P.ADE: .LONG 8
00000008 00000000	00294 P.ADD: .ADDRESS P.ADF
45 4E 4F 2E 47 41 4C 46	00298 P.ADH: .ASCII \FLAG.ONE\
00000008 00000000	002A0 P.ADG: .LONG 8
4C 4C 41 2E 52 45 4C 49	002A4 P.ADJ: .ADDRESS P.ADH
41 52 54 0000000B 00000000	002A8 P.ADJ: .ASCII \TRAILER.ALL\
0000000B 00000000	002B3 P.ADI: .BLKB 1
0000000B 00000000	002B4 P.ADI: .LONG 11
45 4E 4F 2E 52 45 4C 49	002B8 P.ADL: .ADDRESS P.ADJ
41 52 54 0000000B 00000000	002Bc P.ADL: .ASCII \TRAILER.ONE\
0000000B 00000000	002C7 P.ADK: .BLKB 1
54 4E 49 52 50 24 53 4F 43 2E	002C8 P.ADK: .LONG 11
53 49 4C 2E 00000009 00000000	002D0 P.ADM: .ASCII \.COM\
00000009 00000000	002D4 P.ADN: .ASCII \.LIS\
54 4E 49 52 50 24 53 59 53 00000009 00000000	002D8 P.ADP: .ASCII \SYSSPRINT\
00000009 00000000	002E1 P.ADO: .BLKB 3
54 4E 49 52 50 24 53 59 53 00000009 00000000	002E4 P.ADR: .LONG 9
00000009 00000000	002EC P.ADR: .ADDRESS P.ADP
00000009 00000000	002F5 P.ADR: .BLKB 3
47 4F 4C 2E 3A 4E 49 47 4F 4C 24 53 59 53 0000000E 00000000	002F8 P.ADQ: .LONG 9
0000000E 00000000	00300 P.ADT: .ADDRESS P.ADR
0000000E 00000000	0030E P.ADS: .BLKB 2
48 43 54 41 42 24 53 59 53 00000009 00000000	00310 P.ADS: .LONG 14
00000009 00000000	00314 P.ADT: .ADDRESS P.ADT
48 43 54 41 42 24 53 59 53 00000009 00000000	00318 P.ADV: .ASCII \SYSSBATCH\
00000009 00000000	00321 P.ADU: .BLKB 3
5D 4E 5B 20 3F 74 6E 69 72 50 20 2C 53 41 21 3A 00000010 00000000	00324 P.ADU: .LONG 9
00000010 00000000	00328 P.ADX: .ADDRESS P.ADV
5D 4E 5B 20 3F 74 69 6D 62 75 53 20 2C 53 41 21 3A 5D 00000011 00000000	0033C P.ADX: .ASCII \!AS, Print? [N]:\
00000011 00000000	00340 P.ADX: .LONG 16
00000011 00000000	00344 P.ADZ: .ADDRESS P.ADX
00000011 00000000	00353 P.ADZ: .ASCII \!AS, Submit? [N]:\
00000011 00000000	00355 P.ADY: .BLKB 3
00000011 00000000	00358 P.ADY: .LONG 17
00000011 00000000	0035C P.ADY: .ADDRESS P.ADZ

D_\$VERB=	P.AAA
D_P1=	P.AAC
D_AFTER=	P.AAE
D_BURST=	P.AAG
D_CHARACTERISTICS=	P.AAI
D_CLI=	P.AAK
D_COPIES=	P.AAM
D_CPUTIME=	P.AAO
D_DELETE=	P.AAQ
D_DEVICE=	P.AAS
D_FEED=	P.AAU
D_FLAG=	P.AAW
D_FORM=	P.AAY
D_HEADER=	P.ABA
D_HOLD=	P.ABC

D-IDENTIFY=	P.ABE
D-JOB COUNT=	P.ABG
D-KEEP=	P.ABI
D-LOG FILE=	P.ABK
D-LOWERCASE=	P.ABM
D-NAME=	P.ABO
D-NOTE=	P.ABQ
D-NOTIFY=	P.ABS
D-OPERATOR=	P.ABU
D-PAGES=	P.ABW
D-PARAMETERS=	P.ABY
D-PASSALL=	P.ACA
D-PRINTER=	P.ACC
D-PRIORITY=	P.ACE
D-QUEUE=	P.ACG
D-REMOTE=	P.ACJ
D-RESTART=	P.ACK
D-SETUP=	P.ACM
D-SPACE=	P.ACO
D-TRAILER=	P.ACQ
D-USER=	P.ACS
D-WSDEFAULT=	P.ACU
D-WSEXENT=	P.ACW
D-WSQUOTA=	P.ACY
D-BURST_ALL=	P.ADA
D-BURST_ONE=	P.ADC
D-FLAG_ALL=	P.ADE
D-FLAG_ONE=	P.ADG
D-TRAILER_ALL=	P.ADI
D-TRAILER_ONE=	P.ADK
.EXTRN	CLISGET VALUE, CLISPRESENT
.EXTRN	LIBSFIND_FILE, LIBSSCOPY_DDX
.EXTRN	LIBSQUAL_FILE MATCH
.EXTRN	LIBSQUAL_FILE PARSE
.EXTRN	LIBS QUIPRO, LIBS NEGANS
.EXTRN	LIBS FILFAIMAT, PRINTS_FACILITY
.EXTRN	SUBMITS FACILITY
.EXTRN	PRINTS_ADDFIL, SUBMITS_ADDFIL
.EXTRN	PRINTS_CREJOB, SUBMITS_CREJOB
.EXTRN	PRINTS_INVFILSPE
.EXTRN	SUBMITS_INVFILSPE
.EXTRN	PRINTS_REMOINSF
.EXTRN	SUBMITS_REMOINSF
.EXTRN	SUBMITS_INVLOGFIL
.EXTRN	PARSE_AFTER, PARSE CHARACTERISTICS
.EXTRN	PARSE_IF TRUE, PARSE PARAMETERS
.EXTRN	PARSE_PRIORITY, PARSE_IF TRUE_FALSE
.EXTRN	PARSE_USER, PARSE_IF FALSE
.EXTRN	PARSE_QUEUE, PARSE_FORM
.EXTRN	PARSE_COPIES, PARSE_NAME
.EXTRN	PARSE_STRING, PARSE_FILENAME
.EXTRN	PARSE_CPUTIME, PARSE_LOG_FILE
.EXTRN	PARSE_NAME AND LOG FILE
.EXTRN	PARSE_PRINTER, PARSE_WORKING_SET
.EXTRN	SYSSSDNDJBCW, SYSSDCLEXH
.EXTRN	PARSE_LOCAL TRUE FALSE
.EXTRN	PARSE_LOWER_UPPER

.EXTRN PARSE COMMA LIST  
 .EXTRN SYSSOPEN, SYSSCLOSE  
 .EXTRN SYSSCANEXH

			OFFC	00000	SUBMIT:	WORD	SAVE R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	0271	
	57	0000*	CF	9E 00002		MOVAB	FLAGS R7		
	56	FD2D	CF	9E 00007		MOVAB	D DEVICE R6		
	5E	F724	CE	9E 0000C		MOVAB	-2268(SP), SP		
			54	D4 00011		CLRL	CNT		
	E0	6D 0000V	CF	9E 00013		MOVAB	HANDLER (FP)	0298	
		AD 020E0000	8F	D0 00018		MOVL	#34471936, INPUT_DESC	0370	
		E4	AD	D4 00020		CLRL	INPUT_DESC+4	0374	
	E8	AD 020E0000	8F	D0 00023		MOVL	#34471936, VALUE_DESC	0375	
		EC	AD	D4 0002B		CLRL	VALUE_DESC+4	0379	
	D8	AD 020E0000	8F	D0 0002E		MOVL	#34471936, LOG_FILE_DESC	0380	
		DC	AD	D4 00036		CLRL	LOG_FILE_DESC+4	0384	
	F8	AD 020E0000	8F	D0 00039		MOVL	#34471936, FIND_RESULT	0385	
		FC	AD	D4 00041		CLRL	FIND_RESULT+4	0389	
		04	AE	D4 00044		CLRL	FIND_CONTEXT	0390	
	F0	AD 010E0004	8F	D0 00047		MOVL	#17694724, FIND_DEFAULT	0392	
	F4	AD 0238	C6	9E 0004F		MOVAB	P.ADM, FIND_DEFAULT+4	0396	
	FC	A7	01	7D 00055		MOVQ	#1 EXIT STATUS	0397	
		5B	0428	CE 00059		MOVAB	ITEM_BUFFER, Q_ICURSOR	0369	
		5A	28	AE 0005E		MOVAB	DATA_BUFFER, Q_DCURSOR	0408	
		59	00000000*	8F	D0 00062		MOVL	SUBMITS_FACILITY@16>+4904>+4>, -Q_MESSAGE	
			58	E8	AD 9E 00069		MOVAB	VALUE_DESC, Q_VALUE_DESC	
				E8	AD 9F 0006D		PUSHAB	VALUE_DESC	0414
		00000000G	00	FF70	C6 9F 00070		PUSHAB	D_SVERB	
		50	8F	EC	02 FB 00074		CALLS	#2, CLISGET VALUE	
					BD 91 00078		CMPB	QVALUE_DESC+4, #80	0415
					09 12 00080		BNEQ	18	
							BISB2	#1, FLAGS	0418
							MOVAB	P.ADN, FIND_DEFAULT+4	0419
							CALLS	#0, GET_COMM_QUAL_CONTEXT	0424
							PUSHAB	D_IDENTIFY	0429
							CALLS	#T, CLISPRES	
							INSV	RO, #1, #1, FLAGS	
							PUSHAB	D_REMOTE	0430
							CALLS	#T, CLISPRES	
							INSV	RO, #2, #1, FLAGS	
							PUSHAB	D_AFT	0435
							CALLS	#T, PARSE_AFT	
							PUSHAB	D_CHARACTERISTICS	0440
							CALLS	#T, PARSE_CHARACTERISTICS	
							MOVZBL	#71, -(SP)	0446
							PUSHAB	D_HOLD	
							CALLS	#2, PARSE_IF_TRUE	0452
							MOVZBL	#108, -(SP)	
							PUSHAB	D_NOTIFY	
							CALLS	#2, PARSE_IF_TRUE	0457
							PUSHAB	D_PARAMETERS	
							CALLS	#T, PARSE_PARAMETERS	
							PUSHAB	D_PRIORITY	0462
							CALLS	#T, PARSE_PRIORITY	
							MOVZBL	#139, -(SP)	0468
							MOVZBL	#138, -(SP)	

00006 CF	0150	C6 9F 000F2	PUSHAB	D RESTART	0473
00006 CF	018C	03 FB 000F6	CALLS	#3, PARSE_IF_FALSE	
00006 03	01	C6 9F 000FB	PUSHAB	D USER	0476
	67	FB 00104	CALLS	#T PARSE_USER	
	00F6	31 00107	BLBS	FLAGS, 28	
	20	DD 0010A	BRW	5\$	
	01D4	C6 9F 0010C	28:	PUSHL	#32
00006 CF	02	FB 00110	PUSHAB	D BURST_ALL	0482
00006 CF	01E8	21 DD 00115	CALLS	#2, PARSE_IF_TRUE	0483
00006 CF	02	C6 9F 00117	PUSHL	#33	
00006 CF	22	FB 00118	PUSHAB	D BURST_ONE	
00006 CF	9C	DD 00120	CALLS	#2, PARSE_IF_TRUE	0484
00006 CF	02	A6 9F 00122	PUSHAB	#34	
00006 CF	02	FB 00125	CALLS	D BURST	
000000006 00	56	DD 0012A	PUSHL	#2, PARSE_IF_FALSE	0489
10	01	FB 0012C	CALLS	R6	
000000006 00	50	E9 00133	BLBC	#1, CLISPRES	
0F	0130	C6 9F 00136	CALLS	R0, 3\$	
7E	024C	50 E8 00141	PUSHAB	D QUEUE	
86	02	C6 9F 00144	CALLS	#T, CLISPRES	
00006 CF	56	8F 9A 00148	BLBS	R0, 3\$	
	03	DD 0014C	PUSHAB	P. ADO	0491
	24	FB 0014E	PUSHL	#134, -(SP)	
00006 CF	01F8	25 DD 00153	CALLS	R6	
00006 CF	02	C6 9F 00155	PUSHAB	#3, PARSE_QUEUE	0496
00006 CF	25	FB 00159	PUSHL	#36	
00006 CF	0208	DD 0015E	PUSHAB	D FLAG_ALL	
00006 CF	02	C6 9F 00160	PUSHL	#2, PARSE_IF_TRUE	0497
18	02	FB 00164	PUSHAB	#37	
00006 CF	26	DD 00169	PUSHL	D FLAG_ONE	
24	A6 9F 0016B	PUSHAB	#2, PARSE_IF_TRUE	0498	
00006 CF	02	FB 0016E	CALLS	#38	
7E	01	9F 00173	PUSHAB	D FLAG	
48	FB 00176	CALLS	#2, PARSE_IF_FALSE	0503	
64	01	FB 0017B	PUSHAB	D FORM	
00006 CF	64	8F 9A 0017F	CALLS	#T, PARSE FORM	
7E	02	C6 9F 00182	MOVZBL	#75, -(SP)	0509
66	8F 9A 00187	PUSHAB	D JOB_COUNT		
00006 CF	0094	C6 9F 0018B	CALLS	#2, PARSE_COPIES	0515
00006 CF	00A0	02 FB 0018F	MOVZBL	#102, -(SP)	
00006 CF	02	C6 9F 00194	PUSHAB	D LOWERCASE	
7E	01	FB 00198	CALLS	#2, PARSE_IF_TRUE	0520
68	01	FB 0019D	PUSHAB	D NAME	
7E	8F 9A 001A1	CALLS	#T, PARSE NAME		
6A	0F	9A 001A1	MOVZBL	#107, -(SP)	0526
00006 CF	00AC	C6 9F 001A5	PUSHAB	#106, -(SP)	
7E	03	FB 001A9	CALLS	D NOTE	
6F	8F 9A 001AE	MOVZBL	#3, PARSE_STRING	0532	
7E	0F	9A 001B2	PUSHAB	#111, -(SP)	
6E	03	C6 9F 001B6	MOVZBL	#110, -(SP)	
00006 CF	00CC	FB 001BA	PUSHAB	D OPERATOR	
	56	DD 001BF	CALLS	#3, PARSE_STRING	
	01	FB 001C1	PUSHL	R6	
000000006 00	50	E8 001C8	CALLS	#1, CLISPRES	0537
11	0260	C6 9F 001CB	BLBS	R0, 4\$	
7E	86	8F 9A 001CF	PUSHAB	P. ADO	
			MOVZBL	#134, -(SP)	0539

67

0000G CF	0130	C6 9F 001D3	PUSHAB	D_QUEUE	
		03 FB 001D7	CALLS	#3PARSE_QUEUE	
	021C	2B DD 001DC	PUSHL	#43	0544
0000G CF		C6 9F 001DE	PUSHAB	D_TRAILER_ALL	
		02 FB 001E2	CALLS	#2PARSE_IF_TRUE	0545
0000G CF	0230	2C DD 001E7	PUSHL	#44	
		C6 9F 001E9	PUSHAB	D_TRAILER_ONE	
0000G CF		02 FB 001ED	CALLS	#2PARSE_IF_TRUE	0546
0000G CF	0180	2D DD 001F2	PUSHL	#45	
		C6 9F 001F4	PUSHAB	D_TRAILER	
0000G CF		02 FB 001F8	CALLS	#2PARSE_IF_FALSE	0476
		00A9 31 001FD	BRW	78	0554
		12 DD 00200	PUSHL	#18	
		11 DD 00202	PUSHL	#17	
0000G CF	C0	A6 9F 00204	PUSHAB	D_CLI	
		03 FB 00207	CALLS	#3PARSE_FILENAME	0560
0000G CF	E0	16 DD 0020C	PUSHL	#22	
		15 DD 0020E	PUSHL	#21	
0000G CF	7E	A6 9F 00210	PUSHAB	D_CPUTIME	
	60	03 FB 00213	CALLS	#3PARSE_CPUTIME	0566
0000G CF	70	8F 9A 00218	MOVZBL	#96 -(SP)	
0000G CF	58	02 FB 0021C	PUSHAB	D_KEEP	
	D8	AD 9E 00224	CALLS	#2PARSE_IF_TRUE	0571
0000G CF	0080	C6 9F 00228	MOVAB	LOG_FILE_DESC, Q_VALUE_DESC	0572
	03	01 FB 0022C	PUSHAB	D_LOGFILE	
	58	50 F0 00231	CALLS	#TPARSE_LOGFILE	
	E8	AD 9E 00236	INSV	RO, #3, #T, FLAGS	0577
0000G CF	00A0	C6 9F 0023A	MOVAB	VALUE_DESC, Q_VALUE_DESC	0578
	67	01 FB 0023E	PUSHAB	D_NAME	
01		03 E0 00243	CALLS	#TPARSE_NAME	
		8F DD 00247	BBS	#3FLAGS_68	0583
		0278 C6 9F 0024D	PUSHL	#SUBMITS_INVLOGFIL	0587
		D8 AD 9F 00251	PUSHAB	P.ADS	
		E8 AD 9F 00254	PUSHAB	LOGFILE_DESC	
0000G CF	0110	C6 9F 00257	CALLS	#4PARSE_NAME_AND_LOGFILE	0592
0000G CF	028C	C6 9F 00260	PUSHL	D_PRINTER	
	7E	86 8F 9A 00269	CALLS	#TPARSE_PRINTER	0597
0000G CF	0130	C6 9F 0026D	PUSHAB	P.ADU	
	7E	98 8F 9A 00276	CALLS	#134 -(SP)	
0000G CF	97 8F 9A 0027A	PUSHAB	D_QUEUE	#3PARSE_QUEUE	0603
	01A0	C6 9F 0027E	MOVZBL	#152 -(SP)	
0000G CF	9A 8F 9A 00287	PUSHAB	#151 -(SP)		
	7E	03 FB 00282	CALLS	D_WSDEFAULT	
0000G CF	99 8F 9A 00288	MOVZBL	#3PARSE_WORKING_SET	0609	
	01B0	C6 9F 0028F	PUSHAB	#154 -(SP)	
0000G CF	9C 8F 9A 00293	CALLS	#153 -(SP)		
	7E	03 FB 00298	MOVZBL	D_WSEXTENT	0615
0000G CF	98 8F 9A 0029C	PUSHAB	#3PARSE_WORKING_SET		
	01C0	C6 9F 002A0	CALLS	#156 -(SP)	
0000G CF	67	02 E0 002A9	BBS	#155 -(SP)	
		6B D4 002AD	PUSHL	D_WSQUOTA	
		7E 7C 002AF	CLRL	#3PARSE_WORKING_SET	
			CLRO	#2FLAGS_108	
				(Q_CURSOR)	
				-(SP)	

0621  
0631  
0635

00000000G	00	14	AE	9F	00281	PUSHAB	IOSB	0636
	07	0434	CE	9F	00284	PUSHAB	ITEM_BUFFER	
	50	07	7D	00288	MOVQ	#10 -(SP)		
	11	0C	50	E9	002BD	CLRL	-(SP)	
			AE	D0	002C7	CALLS	#7 SYSSNDJBCW	
			50	EB	002CB	BLBC	STATUS, 8\$	0637
			50	DD	002CE	MOVL	IOSB, STATUS	0640
			7E	D4	002D0	BLBS	STATUS, 9\$	0639
						PUSHL	STATUS	
						CLRL	-(SP)	
00000000G	00	00000006	8F	DD	002D2	PUSHL	#SUBMITS CREJOB	0645
		14	03	FB	002D8	CALLS	#3 LIBSSTOP	0646
18	AE	0000V	AE	D4	002DF	CLRL	EXIT_BLOCK	0647
1C	AE		CF	9E	002E2	MOVAB	EXIT_HANDLER, EXIT_BLOCK+4	0648
20	AE	24	01	00	002E8	MOVL	#1 EXIT_BLOCK+8	0649
		24	AE	9E	002EC	MOVAB	EXIT_BLOCK+16, EXIT_BLOCK+12	0650
		14	AE	D4	002F1	CLRL	EXIT_BLOCK+16	
00000000G	00		AE	9F	002F4	PUSHAB	EXIT_BLOCK	
			01	FB	002F7	CALLS	#1 SYSSDCLEXH	0656
		E0	AD	9F	002FE	PUSHAB	INPUT_DESC	
00000000G	00	FF7C	C6	9F	00301	PUSHAB	D P1	
	03		02	FB	00305	CALLS	#2 CLISGET_VALUE	
			50	FB	0030C	BLBS	R0, 11\$	
	5B	0428	0236	31	0030F	BRW	35\$	
	5A	28	CE	9E	00312	11\$:	ITEM_BUFFER, Q_ICURSOR	0664
			AE	9E	00317	MOVAB	DATA_BUFFER, Q_DCURSOR	0665
			18	DD	0031B	PUSHL	#24	0671
0000G	CF	F0	A6	9F	0031D	PUSHAB	D_DELETE	
	03		02	FB	00320	CALLS	#2 PARSE_IF_TRUE	
			67	FB	00325	BLBS	FLAGS, 12\$	0674
		0085	31	00328	BRW	13\$		
			22	DD	0032B	PUSHL	#34	0681
			20	DD	0032D	PUSHAB	#32	
0000G	CF	9C	A6	9F	0032F	CALLS	D_BURST	
			03	FB	00332	PUSHL	#3 PARSE_LOCAL_TRUE_FALSE	
			23	DD	00337	PUSHAB	#35	0687
0000G	CF	D0	A6	9F	00339	CALLS	D_COPIES	
			02	FB	0033C	PUSHAB	#2 PARSE_COPIES	
	7E	76	8F	9A	00341	MOVZBL	#118 -(SP)	0693
	7E	75	8F	9A	00345	MOVZBL	#117 -(SP)	
0000G	CF	0C	A6	9F	00349	PUSHAB	D_FEED	
			03	FB	0034C	CALLS	#3 PARSE_IF_TRUE_FALSE	
			26	DD	00351	PUSHL	#38	0699
			24	DD	00353	PUSHAB	#36	
0000G	CF	18	A6	9F	00355	CALLS	D_FLAG	
	7E	71	03	FB	00358	PUSHAB	#3 PARSE_LOCAL_TRUE_FALSE	
			8F	9A	0035D	MOVZBL	#113 -(SP)	0705
		34	A6	9F	00361	PUSHAB	D_HEADER	
0000G	CF	02	FB	00364	CALLS	#2 PARSE_IF_TRUE		
	7E	5C	8F	9A	00369	MOVZBL	#92 -(SP)	0712
	7E	5B	8F	9A	0036D	PUSHBL	#91 -(SP)	
			2F	DD	00371	PUSHL	#47	
			2E	DD	00373	PUSHAB	#46	
0000G	CF	00DC	C6	9F	00375	CALLS	D_PAGES	
			05	FB	00379	PUSHAB	#5 PARSE_LOWER_UPPER	
	7E	80	8F	9A	0037E	MOVZBL	#128 -(SP)	0718
		0100	C6	9F	00382	PUSHAB	D_PASSALL	

SUBMIT  
V04-000

## PRINT and SUBMIT Commands

12  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55  
VAX-11 BLiss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1Page 26  
(3)

00006 CF	02	FB 00386	CALLS	#2, PARSE_IF_TRUE	0724
	29	DD 0038B	PUSHL	#41	
	28	DD 0038D	PUSHL	#40	
	63	9F 0038F	PUSHAB	D_SETUP	
00006 CF	0160	03 FB 00393	CALLS	#3, PARSE_COMMALIST	0730
	1C	DD 00398	PUSHL	#28	
00006 CF	0170	C6 9F 0039A	PUSHAB	D_SPACE	
	02	FB 0039E	CALLS	#2, PARSE_IF_TRUE	0736
	20	DD 003A3	PUSHL	#45	
00006 CF	0180	C6 9F 003A5	PUSHAB	#43, D_TRAILER	
6E	03	FB 003A7	CALLS	#3, PARSE_LOCAL_TRUE_FALSE	0749
	02	DD 003A8	MOVL	#2, (SP)	
	5E	DD 003B0	PUSHL	SP	
	7E	7C 003B5	CLRL	-(SP)	
	F0	AD 9F 003B7	PUSHAB	FIND_DEFAULT	
	14	AE 9F 003BA	PUSHAB	FIND_CONTEXT	
	F8	AD 9F 003BD	PUSHAB	FIND_RESULT	
	E0	AD 9F 003C0	PUSHAB	INPUT_DESC	
000000006	00	07 FB 003C3	CALLS	#7, LIBSFIND_FILE	
	35	50 DD 003CA	MOVL	RO, STATUS 2	
000182CA	8F	55 D1 003CD	CMPL	STATUS_2, #99018	0750
	52	68 13 003D4	BEQL	17S	
	75	AE DO 003D6	MOVL	FIND_CONTEXT R2	0762
08	AE	55 E9 003DA	BLBC	STATUS 2, 19S	0758
	F8	AD 9E 003DD	MOVAB	FIND_RESULT, PROMPT_DESC	0769
		7E D4 003E2	CLRL	-(SP)	0772
	0C	AE 9F 003E4	PUSHAB	PROMPT_DESC	
	07	67 E9 003E7	BLBC	FLAGS, 14S	0776
	50	02A4 C6 9E 003EA	MOVAB	P. ADW. RO	0777
	50	02C0 C6 9E 003F1	BRB	15S	
		50 DD 003F6	MOVAB	P. ADY. RO	0778
	F8	7E D4 003F8	14S:	RO	
		52 DD 003FA	PUSHL	-(SP)	0772
		A7 9F 003FC	CLRL	R2	0774
000000006	00	06 FB 003FF	PUSHAB	QUAL_CONTEXT	0772
	53	50 DD 00406	CALLS	#6, LIBSQUAL_FILE_MATCH	
	35	53 E8 00409	MOVL	RO, STATUS 3	
000000006	8F	53 D1 0040C	BLBS	STATUS_3, T8S	0783
	20	20 13 00413	CMPL	STATUS_3, #LIBS_FILFAIMAT	0786
000000006	8F	53 D1 00415	BEQL	16S	
	17	17 13 0041C	CMPL	STATUS_3, #LIBS QUIPRO	0787
000000006	8F	53 D1 0041E	BEQL	16S	
	OE	OE 13 00425	CMPL	STATUS_3, #LIBS NEGANS	0788
	F8	AD 9F 00427	BEQL	16S	
		01 DD 0042A	PUSHAB	FIND_RESULT	0791
	00000000*	8F DD 0042C	PUSHL	#1	
000000006	8F	00BD 31 00432	PUSHL	#<<SUBMITS_FACILITY@16>+4920>	0792
		53 D1 00435	BRW	30S	
		35 12 0043C	CMPL	STATUS_3, #LIBS QUIPRO	0795
05 A2	60	FEBD 31 0043E	BNEQ	22S	
		8F 8A 00441	BRW	10S	
000000006	00	52 DD 00446	17S:	#96. 5(R2)	0797
	24	01 FB 00448	PUSHL	R2	0802
	7E	50 F8 0044F	CALLS	#1, SYSSOPEN	0804
	08	A2 7D 00452	BLBS	RO, 23S	
		19S:	MOVO	8(R2), -(SP)	0811

SUBMIT  
VO4-000

## PRINT and SUBMIT Commands

J 12  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55 VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1Page 27  
(3)

			F8	AD	9F	00456	PUSHAB	FIND_RESULT	0806
			08	01	DD	00459	PUSHL	#1	0807
				67	E9	0045B	BLBC	FLAGS, 20\$	0808
			00000000*	06	DD	0045E	PUSHL	#<<<SUBMITS_FACILITY@16>+4248>+2>	
			000000006	00	11	00464	BRB	21\$	
				8F	DD	00466	PUSHL	#<<<SUBMITS_FACILITY@16>+4248>+4>	0809
				05	FB	0046C	CALLS	#5, LIB\$SIG	0807
				3A	31	00473	BRW	13\$	0806
27	14	41	67	A2	02	E1	BBC	#2, FLAGS, 26\$	0814
					05	E0	BBS	#5, 65(R2), 24\$	0817
			F8		AD	0047F	PUSHAB	FIND_RESULT	0819
			000000006	00	01	DD	PUSHL	#1	
				8F	DD	00484	CALLS	#SUBMITS_REMOINSF	
			000000006	00	03	FB	BRB	#3, LIB\$SIG	
				0E	11	00491	BLBC	26\$	
			05	A2	67	E9	BISB2	FLAGS, 25\$	0823
					20	88	BRB	#32, 5(R2)	0824
			05	A2	05	11	BISB2	26\$	
					8F	88	CALLS	#64, 5(R2)	0825
			000000006	00	52	DD	PUSHL	R2	0829
				16	01	FB	BLBS	#1, SYSCLOSE	
				7E	50	E8	MOVQ	R0, 27\$	
					08	A2	PUSHAB	8(R2), -(SP)	0834
			F8		AD	9F	PUSHL	FIND_RESULT	0831
			000000006	00	01	DD	CALLS	#1	
				8F	DD	004B6	PUSHL	#<<<SUBMITS_FACILITY@16>+4176>+2>	0832
AC	OD	41	67	A2	05	FB	CALLS	#5, LIB\$SIG	0837
					02	E0	BBS	#2, FLAGS, 22\$	0840
			F8		05	E1	BBC	#5, 65(R2), 28\$	0842
			000000006	00	01	DD	PUSHAB	FIND_RESULT	
				8F	DD	004C3	PUSHL	#1	
					02	E0	BRB	#SUBMITS_REMOINSF	
					05	E1	BLBS	30\$	
				F8		AD	BBC	#6, 65(R2), 29\$	0847
			000000006	00	01	DD	PUSHL	67(R2), 29\$	0848
				8F	DD	004C7	CALLS	#5, 64(R2), 31\$	0849
					05	E1	BRB	FIND_RESULT	0851
				F8		AD	PUSHL	#1	
			000000006	00	19	11	BRB	#SUBMITS_INVFILSPE	
				8F	DD	004D1	PUSHL	#3, LIB\$SIG	
09	41	A2	06	E1	004D9	28\$:	CALLS	34\$	
			43	A2	E8	004DE	BRB	40(R2), NAM	0864
14	40	A2	05	E1	004E2	29\$:	MOVL	#2555932 (Q ICURSOR)	0866
			F8		AD	9F	MOVL	20(R0), 4(Q ICURSOR)	0868
			000000006	00	01	DD	MOVAB	B(Q ICURSOR)	0869
				8F	DD	004EC	CLRQ	-(SP)	0878
					03	FB	CLRQ	IOSB	
					4A	11	PUSHAB	ITEM_BUFFER	
					14	004F9	PUSHAB	#2, -(SP)	
				50	A2	DD	MOVQ	-(SP)	
			28	8F	DD	004FB	CLRL	#7, SYSSNDJBCW	0879
			6B 0027001C		A0	9E	CALLS	STATUS, 32\$	
04	AB		14			00506	BLBC	IOSB, STATUS	0880
			08		AB	7C	MOVAB	STATUS, 33\$	0884
					7E	7C	CLRQ	STATUS	0882
					14	9F	PUSHAB	FIND_RESULT	
			0434		CE	00513	PUSHAB	#1	
			7E		02	7D	MOVQ		
			000000006	00	07	FB	CLRL		
				07		0051C	CALLS		
				50		E9	BLBC		
				50		00523	MOVAB		
				50		AE	MOVAB		
				50		D0	CLRL		
				50		E8	CALLS		
				50		0052A	BLBC		
				50		DD	MOVAB		
				50		0052D	MOVAB		
				50		9F	CLRL		
				50		00532	CALLS		
				01		DD	BLBC		

SUBMIT  
VO4-000

## PRINT and SUBMIT Commands

11

16-Sep-1984 00:28:51  
14-Sep-1984 12:09:59

16-Sep-1984 00:28:51 VAX-11 BLiss-32 V4.0-742  
14-Sep-1984 12:09:55 [CLIUTL.SRC]SUBMIT.B32:1

Page 28 (3)

; Routine Size: 1504 bytes, Routine Base: CODE + 0360

```

854 0960 1 ROUTINE GET_COMM_QUAL_CONTEXT: NOVALUE =
855 0961 1 ++
856 0962 1
857 0963 1 FUNCTIONAL DESCRIPTION:
858 0964 1 This routine is gets the common qualifier context.
859 0965 1
860 0966 1 INPUT PARAMETERS:
861 0967 1 NONE
862 0968 1
863 0969 1 IMPLICIT INPUTS:
864 0970 1 QUAL_CONTEXT - defined locally in main routine.
865 0971 1
866 0972 1 OUTPUT PARAMETERS:
867 0973 1 NONE
868 0974 1
869 0975 1 IMPLICIT OUTPUTS:
870 0976 1 NONE
871 0977 1
872 0978 1 ROUTINE VALUE:
873 0979 1 NONE
874 0980 1
875 0981 1 SIDE EFFECTS:
876 0982 1 NONE
877 0983 1
878 0984 1 !--
879 0985 1
880 0986 2 BEGIN
881 0987 2
882 0988 2 LOCAL
883 0989 2 BITMAP,
884 0990 2 STATUS;
885 0991 2
886 0992 2 ! Set up the context bitmap
887 0993 2
888 0994 2 BITMAP= LIBSM_CQF_CONFIRM OR LIBSM_CQF_EXCLUDE OR
889 0995 2 LIBSM_CQF_BEFORE OR LIBSM_CQF_SINCE OR
890 0996 2 LIBSM_CQF_CREATED OR LIBSM_CQF_MODIFIED OR
891 0997 2 LIBSM_CQF_EXPIRED OR LIBSM_CQF_BACKUP OR
892 0998 2 LIBSM_CQF_BYOWNER;
893 0999 2
894 1000 2 ! Get the common qualifier context
895 1001 2
896 1002 3 IF NOT (STATUS = LIBSQUAL_FILE_PARSE (BITMAP, QUAL_CONTEXT))
897 1003 2 THEN
898 1004 2 SIGNAL_STOP (.STATUS);
899 1005 2
900 1006 1 END;

```

0000 00000 GET\_COMM\_QUAL\_CONTEXT:

7E	01FF	8F 3C 00002	WORD Save nothing
	0000	CF 9F 00007	MOVZWL #511, BITMAP
	04	AE 9F 00008	PUSHAB QUAL_CONTEXT
			PUSHAB BITMAP

0960  
0997  
1002

SUBMIT  
V04-000

PRINT and SUBMIT Commands

M 12  
16-Sep-1984 00:28:51    VAX-11 Bliss-32 v4.0-742  
14-Sep-1984 12:09:55    [CLIUTL.SRC]SUBMIT.B32;1

Page 30  
(4)

00000000G 00	02	FB 0000E	CALLS	#2, LIB\$QUAL_FILE_PARSE
09	50	E8 00015	BLBS	STATUS, 1\$
00000000G 00	50	DD 00018	PUSHL	STATUS
	01	FB 0001A	CALLS	#1, LIB\$STOP
	04	00021 18:	RET	

1004  
1006

; Routine Size: 34 bytes.    Routine Base: CODE + 0940

```

902 1007 1 ROUTINE EXIT_HANDLER: NOVALUE=
903 1008 1 ++
904 1009 1
905 1010 1
906 1011 1 FUNCTIONAL DESCRIPTION:
907 1012 1 This routine is an exit handler that is established as long as the
908 1013 1 process owns an open job. Its function is to delete the open job
909 1014 1 if an image exit should occur.
910 1015 1
911 1016 1 INPUT PARAMETERS:
912 1017 1 Standard exit handler parameters (not used).
913 1018 1
914 1019 1 IMPLICIT INPUTS:
915 1020 1 NONE
916 1021 1
917 1022 1 OUTPUT PARAMETERS:
918 1023 1 NONE
919 1024 1
920 1025 1 IMPLICIT OUTPUTS:
921 1026 1 NONE
922 1027 1
923 1028 1 ROUTINE VALUE:
924 1029 1 NONE
925 1030 1
926 1031 1 SIDE EFFECTS:
927 1032 1 NONE
928 1033 1
929 1034 1 !--
930 1035 1
931 1036 2 BEGIN
932 1037 2 $SNDJBC(FUNC=SJCS_CLOSE_DELETE);
933 1038 1 END;

```

.EXTRN SYSSNDJBC

0000 00000 EXIT_HANDLER:			
	7E	7C 00002	.WORD CLRQ Save nothing
	7E	7C 00004	-(SP)
7E	08	7D 00006	CLRQ -(SP)
00000000G 00	7E	D4 00009	MOVQ #8, -(SP)
	07	FB 0000B	CLRL -(SP)
	04	00012	CALLS #7, SYSSNDJBC
			RET

: Routine Size: 19 bytes. Routine Base: CODE + 0962

1007  
1037

1038

```
935 1039 1 ROUTINE HANDLER(SIG,MECH)=
936 1040 1
937 1041 1 ++
938 1042 1
939 1043 1 FUNCTIONAL DESCRIPTION:
940 1044 1 This routine is the condition handler for the main routine. It
941 1045 1 saves the most severe condition as the exit status.
942 1046 1
943 1047 1 INPUT PARAMETERS:
944 1048 1 Standard VMS condition handler parameters.
945 1049 1
946 1050 1 IMPLICIT INPUTS:
947 1051 1 NONE
948 1052 1
949 1053 1 OUTPUT PARAMETERS:
950 1054 1 NONE
951 1055 1
952 1056 1 IMPLICIT OUTPUTS:
953 1057 1 EXIT_STATUS - Contains the most severe status encountered.
954 1058 1
955 1059 1 ROUTINE VALUE:
956 1060 1 SSS_RESIGNAL
957 1061 1
958 1062 1 SIDE EFFECTS:
959 1063 1 NONE
960 1064 1
961 1065 1 --
962 1066 1
963 1067 2 BEGIN
964 1068 2 MAP
965 1069 2 SIG: REF BBLOCK. ! Signal parameters
966 1070 2 MECH: REF BBLOCK; ! Mechanism parameters
967 1071 2 BIND
968 1072 2 SIGNAME = SIG[CHFSL_SIG_NAME] : BBLOCK; ! Name of signal
969 1073 2
970 1074 2
971 1075 2 ! Change the facility code from SUBMIT to PRINT if required.
972 1076 2
973 1077 2 IF
974 1078 2 .FLAGS[V_PRINT]
975 1079 2 AND NOT .SIGNAME
976 1080 2 AND .SIGNAME[STSSV_FAC_NO] EQL SUBMITS_FACILITY
977 1081 2 THEN
978 1082 2 SIGNAME[STSSV_FAC_NO] = PRINTS_FACILITY;
979 1083 2
980 1084 2
981 1085 2 ! Save the most severe condition as the exit status.
982 1086 2
983 1087 2 IF
984 1088 2 NOT .SIGNAME
985 1089 4 AND {(.SIGNAME[STSSV_SEVERITY]
986 1090 4 GTRU .EXIT_STATUS[STSSV_SEVERITY])
987 1091 3 OR .EXIT_STATUS[STSSV_SEVERITY])
988 1092 2 THEN
989 1093 2 EXIT_STATUS = .SIGNAME;
990 1094 2
991 1095 2
```

992 1096 2 SSS RESIGNAL  
993 1097 1 END:

! Resignal to get message

; Routine Size: 68 bytes, Routine Base: CODE + 0975

SUBMIT  
V04-000

PRINT and SUBMIT Commands

D 13  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55  
VAX-11 Bliss-32 v4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 34  
(7)

: 995 1098 1 END  
: 996 1099 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
DATA CODE	12 NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2) 2489 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	

Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	107	1	581	00:00.9

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SUBMIT/OBJ=OBJ\$:SUBMIT MSRC\$:SUBMIT/UPDATE=(ENH\$:SUBMIT)

: Size: 1625 code + 876 data bytes  
: Run Time: 00:32.4  
: Elapsed Time: 01:47.3  
: Lines/CPU Min: 2037  
: Lexemes/CPU-Min: 19676  
: Memory Used: 440 pages  
: Compilation Complete

0059 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

